# Multi-channel Key Agreement using Encrypted Public Key Exchange

Bruce Christianson and Jun Li

School of Computer Science, University of Hertfordshire
College Lane, Hatfield AL10 9AB, England, Europe

**Abstract.** We present a new protocol for cryptographic key agreement between devices which have had no previous association, and which does not rely upon mutual access to a pre-existing key infrastructure. This protocol is suitable for use in mobile ad-hoc computing environments, where the only channels with high data origin authenticity have severely limited bandwidth. The protocol illustrates one use of an heretical design principle: allowing the "same" protocol to provide different security services in different contexts.

## 1 Introduction

In ubiquitous computing [1], ad-hoc sessions must frequently be initiated between devices such as Personal Digital Assistants (PDAs). There is sometimes a need for such sessions to be secured by a cryptographic key, possibly to ensure confidentiality, but usually more importantly to ensure data integrity and originator authenticity. The devices and their owners may have had no previous contact or association, and there is in reality no guarantee of on-line access to a suitably mutually trusted Public Key Infrastructure (PKI). In any case such infrastructures currently address subtly the wrong security requirement: in the ubiquitous context, the primary objective of the participants is not to learn or validate the identity of the other party to whom they are speaking, but is rather to establish secure communication between their own PDA and a PDA being held by the person whom they already know to be the "correct stranger" [2].

To do this, the two PDAs must somehow agree a fresh strong cryptographic key, but must do this by exchanging messages only over public channels where the information which they exchange can be overheard and possibly altered. At the end of this protocol, the owners must be justified in believing that the new key has been shared between the correct pair of PDAs, and is not known to any other device or person.

A classical solution to this key-agreement problem is Diffie-Hellman (DH) key exchange [3]. However conventional DH relies upon the existence of a high bandwidth channel with high data origin authenticity, a combination of properties which is not generally available in the ubiquitous computing scenario.

## 2 Multi-Channel Protocols

In this ubiquitous context, there is increasing interest in "multi-channel" security protocols [1, 4–6], in which we explicitly model different channels, with different characteristics, over which the devices may communicate. In what follows we shall assume a scenario with two channels having the following characteristics:

Channel one is a relatively high bandwidth channel, which is subject to both passive and active attacks [7], including message deletion, insertion and alteration, masquerade and man in the middle. We can think of channel one as being realised by an RF connection.

Channel two is a relatively low bandwidth channel which is subject to passive attack (eavesdropping) but not to active attack, and which has high data origin authenticity: the owner of each device is assured that a message on this channel really does come from the other device.

We can think of the second channel as being realised by one device displaying a number on the display, and the owner of the second device typing this number into their keypad [6]. The second channel could alternatively be realised by physical contact between the devices [4], by an optical channel such as an infrared link ubiq, by one device playing a tune which is recorded by the other, or displaying a bar code which can be photographed and decoded by the other [5], and so on. It is assumed that transmitting more than (say) 40 bits in each direction on channel two during a protocol run will be onerous in time and inconvenient for the humans.

It is important to note that, as far as the threat model is concerned, the endpoints of the second channel are the Application Program Interfaces (APIs) to the cryptographic modules inside the PDAs, not the PDA user interfaces such as screen and keyboard. This observation about channel endpoints is particularly significant in case the high-level security requirement is for integrity rather than confidentiality.

The requirement for data origin authenticity on the second channel therefore entails instantiation of some unspoofable mechanism (such as a red light) which provides the human user with the necessary assurance that the PDA keyboard and display are indeed internally connected to the relevant crypto-module API during the times when messages are being passed on the second channel.

We shall not assume that it is possible for the owner of the PDA to key a "secret" such as a PIN into the keypad, or to read a number on the display, without being overlooked. The attacker may also be able to exploit spyware running inside the PDA to view (but not to modify) messages sent over the second channel. However it is assumed that the attacker cannot see what is going on inside the crypto-module itself. In particular we assume that the crypto-module can generate (or at least access) random numbers which are invisible to the attacker.

## 3 The New Protocol

The human users Alice and Bob have control of devices $A$ and $B$ respectively. The Diffie-Hellman (DH) modulus $g$ and base $q$ can be publicly known integer values, which are independent of both device and person identities and so can be pre-loaded into the devices, or else they may be agreed ad hoc.

Devices $A$ and $B$ pick strong secrets (Kilobit entropy) $x$ and $y$ respectively, and weak (20-40 bit entropy) secrets $k_A$ and $k_B$ respectively. Define $z = g^{xy} \bmod q$, and break $z$ up into fields, so that

$$z = c|s|n_A|n_B$$

where | denotes concatenation. For example we may define $n_B$ to be the first 50 bits of $z$, starting the count from the least significant bit, $n_A$ to be the next 50 bits of $z$, and the session key $s$ to be the next 250 bits, but these bit lengths are configuration constants.

The notation $A \longrightarrow B : X$ means $A$ transmits the message $X$ to $B$ over channel 1, and $A \Longrightarrow B : X$ means $A$ transmits the message $X$ to $B$ over channel 2. The protocol is as follows:

$$A \longrightarrow B : g^x + k_A \bmod q \tag{1}$$
$$B \Longrightarrow A : OK \tag{2}$$
$$B \longrightarrow A : g^y + k_B \bmod q \tag{3}$$
$$A \Longrightarrow B : OK, k_A \tag{4}$$
$$B \Longrightarrow A : k_B \tag{5}$$
$$A \longrightarrow B : n_A \tag{6}$$
$$B \longrightarrow A : n_B \tag{7}$$

$A$ and $B$ check that the received value of $n_B$ or $n_A$ respectively matches that obtained from their calculation of $z$, and announce successful completion of the protocol if it does. If the protocol runs correctly to completion, then Alice and Bob can each be sure that $s$ is a secret shared between $A$ and $B$, and not shared with any attacking device.

We assume that all calculations involving secrets take place inside the crypto-modules, and only the values specified in the protocol messages are allowed to leave the crypto-modules. In particular $x$, $y$ and $s$ never leave. The value calculated by $B$ for $n_A$ must be concealed until the penultimate message has been received, and similarly for $A$ and $n_B$ with the final message.

## 4 Discussion

The innovative new feature of our protocol is to enhance conventional DH key exchange by super-encyphering the public keys $g^x \bmod q$ and $g^y \bmod q$ with the weak secret keys $k_A$ and $k_B$ respectively. Once the devices confirm, via

the second channel, that they have committed to the super-encyphered values received over the first channel, these weak keys are then revealed, again via the second channel. The final pair of messages, which are not bit-limited, ensure that the correct super-encyphered values were in fact received, and hence that the devices have the same value for the key $s$.

The attacker cannot solve the discrete logarithm problem, and so cannot obtain any bits of $z$. Nor can the attacker successfully masquerade as one of the participating PDAs by interposing on the first channel a value for which the attacker knows the exponent, since he is forced to commit to a super-encypherment of this value before he learns the value of the weak key which will be used to decypher it.

Conceptually, conventional DH key exchange over a single channel consists of the messages sent over channel one in our protocol, with $k_A$ and $k_B$ set to zero. However the two messages from $B$ to $A$ are usually combined for convenience in the DH protocol, whereas in our protocol they must be separated in order to avoid premature revelation of $k_B$.

Our approach can also be regarded as a variation of an Encrypted Key Exchange (EKE) protocol [8, 9], but the weak keys used here are short-term, initially unshared, and publicly revealed; whereas traditional EKE uses pre-shared long-term secrets as weak keys. A protocol similar to ours here is also given in [2], but there it provides a very different security service, under different assumptions.

The protocol in the present paper achieves similar objectives to those discussed in [6], but makes considerably more effective use of the bandwidth of the second channel: twenty bits in each direction reduces the attacker's chance of success to less than one in a million, regardless of how much pre-computation the attacker is prepared to invest. Another major difference with [6] is that our protocol requires the data transferred via the second channel to be used in calculating subsequent protocol values, rather than merely to be checked for equality. This is a virtue: humans may check carelessly if there is no immediate reason to be conscientious, and the labour of transferring 20 bits, even with added Hamming redundancy, is still less than that of entering a single telephone number.

## 5 Generalizations

The protocol given here can readily be generalised to a multi-party case, similar to the context of the protocols considered in [10]. The analysis and precautions given in [9] can also readily be applied to our new protocol: in particular, ElGamal [11] can be used in place of DH if transfer of conventional public key certificates is required, see [9] for details.

The protocol given here can also, perhaps surprisingly, be used almost unchanged in a time-limited context. Here the security requirement is to prove physical proximity of the devices initiating a session, and the second channel

is a time-limited channel, usually realised (at a lower level of abstraction) by a bitwise challenge-response protocol [12].

In our setting the participants have no reliable means to pre-share information, so it is convenient to combine the exchange of $k_A$ and $k_B$ over the second channel as follows. Let $a[i], b[i]$ denote the $i$-th bit of $k_A, k_B$ respectively, and let $d[0]$ be a one-bit challenge chosen at random by $B$. The challenge-response sequence proceeds as follows:

$$B \Longrightarrow A : d[0] \tag{8}$$
$$A \Longrightarrow B : c[1] = d[0] \oplus a[1] \tag{9}$$
$$B \Longrightarrow A : d[1] = c[1] \oplus b[1] \tag{10}$$
$$A \Longrightarrow B : c[2] = d[1] \oplus a[2] \tag{11}$$
$$\cdots \tag{12}$$
$$A \Longrightarrow B : c[n] = d[n-1] \oplus a[n] \tag{13}$$
$$B \Longrightarrow A : d[n] = c[n] \oplus b[n] \tag{14}$$

In this sequence, each response is used as the unpredictable time-limited challenge to the other party for the next response in the other direction. This means that, for a given level of security, the time limited channel need convey only one more bit than in the original protocol[1] As in the previous case, a precisely limited amount of Forward Error Correction can be incorporated into the keys exchanged over the second channel; alternatively the correct values can immediately be exchanged over the first channel. In either case errors above the acceptable threshold level for the second channel cause the protocol run to be aborted.

The protocol illustrates the use of an heretical design principle: allowing the "same" protocol to provide different security services in different contexts. It is fascinating to speculate whether, using such protocols, PDAs could become involved in multi-channel interactions with devices like cash points and credit card readers in such a way as to reduce the possibilities for fraud.

## References

1. Feeney, L.M., Ahlgren, B., and Westerlund, A.: Spontaneous Networking: An Application-Oriented Approach to Ad-Hoc Networking, IEEE Communications Magazine, 39(6), 176–181 (June 2001)
2. Li, J., Christianson, B., and Loomes, M.: "Fair" Authentication in Pervasive Computing, In: Secure Mobile Ad-hoc Networks and Sensors (Proc MADNES05), LNCS, vol. 4074, pp 132–143, Springer (2006)

[1] The conventional distance-bounding approach requires $4n$ bits to be exchanged over the time-bounded channel for $n$ bits of security: $2n$ challenges and $2n$ responses. For each of these pairs, it suffices for the man-in-the-middle to guess the correct value for *either* the challenge *or* the response.

3. Diffie, W., and Hellman, M.: New Directions in Cryptography, IEEE Trans Info Theory, 22(6), 644–654 (1976)
4. Stajano, F., and Anderson, R.: The Resurrecting Duckling: Security Issues for Ad-Hoc Wireless Networks, Security Protocols 7, LNCS, vol. 1796, pp 172–194, Springer (2000)
5. McClune, J.M., Perrig, A., and Reiter, M.K.: Seeing-is-Believing: Using Camera Phones for Human-Verifiable Authentication, In: Proc IEEE Security and Privacy, Oakland 2005, 101–124 (2005)
6. Wong, F-L., and Stajano, F.: Multi-Channel Protocols: Strong Authentication using Camera-Equipped Wireless Devices, In: Security Protocols 13, LNCS, vol. 4631, pp 112–132, Springer (2007)
7. Stallings, W.: Cryptography and Network Security, 4ed, Pearson Prentice Hall (2006)
8. Bellovin, S.M., and Merritt, M.: Encrypted Key Exchange: Password-Based Protocols Secure against Dictionary Attacks, In: Proc IEEE Security and Privacy, Oakland 1992, 72–84 (1992)
9. Christianson, B., Roe, M., and Wheeler, D.: Secure Sessions from Weak Secrets, In: Security Protocols 11, LNCS, vol. 2264, pp 190–212, Springer (2005)
10. Creese, S.J., Goldsmith, M.H., Roscoe, A.W., and Xiao, M.: Bootstrapping Multi-Party Ad-Hoc Security, In: Proc ACM SAC 2006, 369–375 (2006)
11. ElGamal, T.: A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms, IEEE Trans Info Theory, 31(4), 469–472 (1985)
12. Clulow, J., Hancke, G.P., Kuhn, M.G., Moore, T.: So Near and Yet So Far: Distance-Bounding Attacks in Wireless Networks, In: ESAS 2006, LNCS, vol. 4357, pp 83–97 (2006)